大规模多核处理器测试技术

齐子初 刘慧

摘 要：片上多核处理器的发展使得片上晶体管的数量成倍剧增，为测试带来了巨大的挑战。最显著的问题集中在测试向量急剧增长、读写（I/O）资源限制更为明显、测试功耗成倍增长、实速测试更为复杂，这些都导致了测试成本的急剧增长。本文系统研究了片上多核处理器测试中的关键问题，重点介绍了基于同构核的测试访问和测试控制机制设计技术。

关键词：微处理器测试 多核微处理器测试

1 引言

随着芯片工艺不断向纳米级的方向发展，芯片的规模越来越大，芯片的测试面临越来越多的挑战，其中测试成本就是需要考虑的重要挑战之一。测试成本不断升高的一个重要因素是测试向量的急剧增加导致的测试时间的增加。由于测试需要的数据量与电路复杂度呈指数增长关系，在高性能微处理器测试中，扫描测试的向量常常达到了千兆字节。ITRS[1]2007[1]报告预测，CPU 包含的晶体管数目在 2015 年会达到 24 亿，而且这个数据还在不断更新。随着芯片制造工艺的发展，单个晶体管的制造成本不断下降，而单个晶体管的测试成本基本保持不变，甚至由于纳米级工艺下新故障的出现，单个晶体管的测试成本还有缓慢上升的趋势。大量的测试向量导致了测试时间和测试成本的迅猛增加。

为了最大程度地提高多核系统测试的并行度，以降低测试时间和提高测试质量，本文系统地介绍了片上多核处理器测试中的关键问题所在，并重点介绍了基于同构核的测试访问和测试控制机制设计技术。

2 背景

片上多核处理器和众多的片上系统（SoC）芯片在设计时有某些共同之处，即采用了基于核的设计以及核复用的技术。基于核的设计提高了芯片的设计效率，缩短了芯片设计的周期。但随着芯片内芯核的增多，芯片的测试变得越来越困难。由于对芯片内所有的核很难同时测试，只能分批顺序地进行，因此就需要一种测试控制策略和适当的测试通道划分机制来控制对芯片内的核的测试。基于 IP 核的测试设计主要集中在三个方面：核的封装器(Wrapper)设计；测试访问机制（TAM，Test Access Mechanism）设计；基于核的测试调度优化。核测试封装可以对 IP 核进行隔离，并提供 IP 核与它所在的测试环境之间的接口和提供 IP 核的多种测试模式。由于 IP 核嵌入在芯片的内部，其输出输入端口不能像芯片的输入输出引脚那样直接访问，所以在某种程度上降低了 IP 核的输入和输出端口的可控制性和可观测性。在 IP 核封装设计中，需要控制和观测 IP 核的输入和输出端口，将核输入和输出端口与片上其他逻辑相隔离，使得被测 IP 与其他逻辑不相互干扰，不受其他的核的影响，以达到单独进行测试的目的。测试访问机制的作用是传输测试数据，把芯片外部的测试激励传输至被测试核的端口或者扫描链上，并把测试响应从被测试核的端口或者扫描链传输出去。这三者有

[1] The International Technology Roadmap for Semiconductors，国际半导体技术路线图

机地结合起来，能够最大程度地降低复杂多核芯片的测试时间和测试开销。

2.1 封装器的设计技术

为了满足对嵌入式内核的测试需求，TTTC 在 1997 年专门成立了嵌入式内核标准制定（Standard for Embedded Core Test）工作组，于 2005 年制定了 IEEE 1500 标准[2]。IEEE1500 利用 WBR（封装边界寄存器，Wrapper Boundary Register）对核的输入输出端口施加测试激励和捕获测试响应。WBR 由类似于 IEEE1149.1 的边界扫描单元的单个 WBC[2]组成。WBR 由 4 种功能模式组成：正常模式（Normal mode）、内部面向模式（Inward Facing Mode）、外部面向模式（Outward Facing Mode）和安全模式（Safe Mode）。当处于正常模式下，WBR 对于系统和核来说是透明的，此时核运行于正常功能模式下；当处于内部面向模式时，核的输入由 WBR 控制，而输出通过 WBR 来观测，在这一状态下对核内部逻辑测试；当处于外部面向模式时，核的输出由 WBR 控制而输入由 WBR 观测，在这一状态下对核外部逻辑测试；当处于安全模式时，核的输入核和输出处于一种固定的状态。

为了方便地测试核间的胶合逻辑、控制电路和核内部的电路，IEEE1500 提供了几种主要的测试模式：正常功能模式、INTEST 测试模式、EXTEST 测试模式和旁路测试模式。正常功能模式下，核的输入输出端口不受测试模式的控制，所有的核处在正常功能运行的状态。在 INTEST 模式时，测试的是核内的逻辑，核内的激励和响应都通过 WBR 移入和移出。此时 WBR 处于一种内部面向模式，即核内的激励和响应是由 WBR 来控制和观测的。在 EXTEST 模式时，测试的是核间的连线和胶合逻辑，核外的激励和响应也通过 WBR 移入和移出，但 WBR 处于一种外部面向模式，即核的输出和核外的响应分别由 WBR 控制和观测。在处于旁路模式时，核内处于功能模式，和 IEEE1149.1 类似，核内的测试通道是直接被旁路的，这样测试数据可以快速地穿过非测试核。图 1 给出了一个 IEEE1500 测试外壳的结构图。

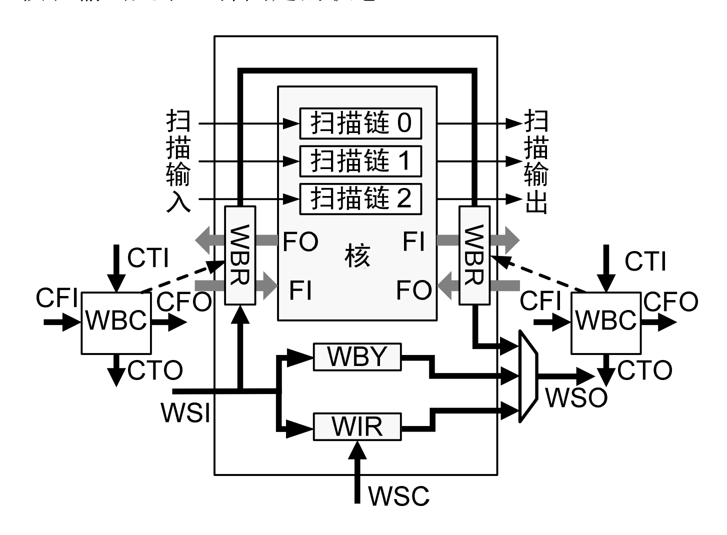


图1. IEEE1500 结构

图中：
WBY: Wrapper Bypass Register，旁路寄存器
WIR: Wrapper Instruction Register，指令寄存器
WSI/WSO: Wrapper Serial Input/Output，串行输入/输出接口
CFI/CFO: Cell Function Input/Output,单元功能输入/输出接口
CTI/CTO: Cell Test Input/Output，单元测试输入/输出接口

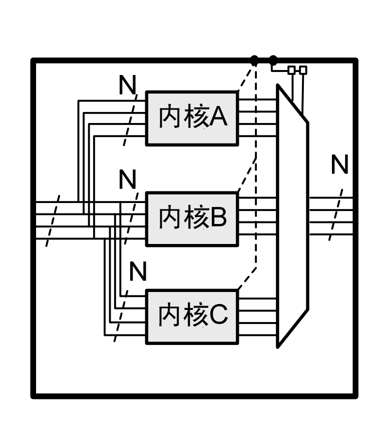
和 IEEE1149.1 类似，IEEE1500 本身只对 IP 核的隔离机制、核的测试模式进行了规范，并不对芯片内部的测试方法或者可测性设计进行标准化，也不对系统级的测试访问机制进行规定。IP 核本身的测试架构是芯核设计者必须要考虑的问题，而系统级的测试访问机制由片上系统的集成者或者多核芯片的设计者解决。

2.2 测试访问机制的设计技术

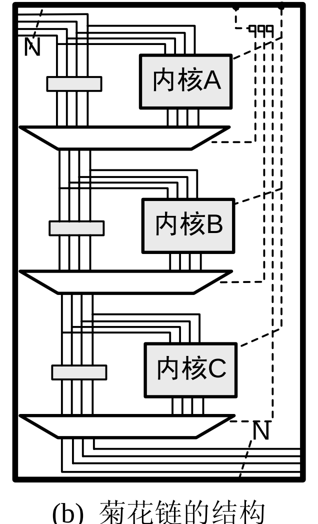
在多核芯片测试时，测试访问机制的作用是传输 IP 核的测试数据，即把测试激励从芯片外部传输至被测试核的外壳，并把测试响应从测试外壳传输至芯片外部。目前在多核芯片

[2] Wrapper Boundary Cell，边界暂存器

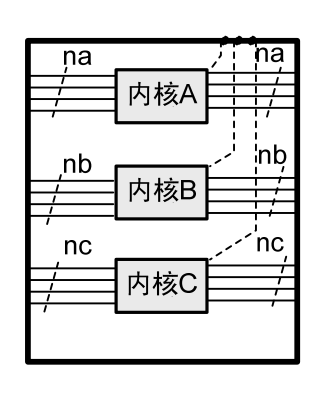
的测试设计中，测试访问机制主要有三种基本的访问方式：多路选择器（Multiplexed）结构、菊花链（Daisy-chained）结构、分布式的（Distribution-based） 结构[3]，如图 2 所示。



(a) 多路选择器结构



(b) 菊花链的结构



(c) 分布式的结构

图2. 传统的测试访问机制结构

在多路选择器的测试访问机制中，每个 IP 核的测试输入都直接连接到芯片的测试输入上，而每个 IP 核的测试输出通过选择器选择输出到芯片的测试输出通道。测试通道的数目等于测试通道数最多的 IP 核的数目。在多路选择器结构中，任何时候只有一个 IP 核被测试，整个测试只能串行完成，因此测试时间比较长，而且核间的逻辑很难被测到。

在菊花链的测试访问机制中，所有的核都可以访问所有的测试通道，可以顺序对每个核进行测试。为了缩短每个核的测试输入延迟，对每个核都引入了旁路存储器，并用一个多路选择器来选择是前一个核的测试输出数据还是旁路的测试通道数据。当部分核并行测试时，必须引入一个旁路的测试输入与输出控制逻辑。这个旁路逻辑设计起来比较复杂，而且对于有很多个核的芯片来说，由于多路选择器的级数比较多，会使得测试输入时间明显增加。

在分布式的测试访问机制结构中，每个核都有固定的测试通道，每个核的通道数目根据被测试核的不同而有所不同。但所有核的测试通道数目之和等于总的测试通道数目。由于各个 IP 核在测试的时候没有测试通道的冲突，利用这种结构可以很方便地对多个核进行并行测试。但随着芯片内核数目的增加，所需要的测试通道数目也需随之增加，由于芯片的读写端口的数目有限，很难做到分配给每个核足够的测试通道。

上述三种测试访问机制结构都有其特定的优点和缺点，在测试设计中通常采用集合一种或者多种的测试访问机制结构。例如专用测试总线 Test Bus[4]集合了多路选择器的测试访问机制和分布式的测试访问机制结构，而 Test Rail[5]既使用了菊花链的测试访问机制结构也使用了分布式的结构。也有其他的测试访问机制设计策略，例如核透明化策略[6]，隔离环[7]，以及总线策略等。其中采用总线策略的包括复用系统总线的 AMBA[8]，以及 CAS（Core Access Switch）[9]等。

3 现有多核处理器的测试控制和测试访问机制设计

多核处理器测试发展的一个趋势是对多个同构核和整体芯片隔离，并分开进行测试。这就需要对同构处理器核进行测试封装和对整体芯片进行测试分割并进行并行测试。对同构核处理器进行并行测试的一个简单方法就是对每个处理器核都分配固定的测试通道来完成对

所有核的同时测试。比较典型的有 Cell 处理器[10]、UltraSPARC T1 处理器[11]、UltraSPARC Niagara2 处理器[12]等。这种方法可以很快捷地得到每个核的测试结果。但随着多核处理器中核数目的增多、规模的增大，有限的测试通道资源使得扫描链的长度越来越长，大大增加了测试的时间。为了降低测试时间，有些处理器采用了更为有效的测试控制和测试访问机制。其中最典型的设计有 AMD Opteron 4 核芯片[13]、UltraSPARC 2 核芯片[14]和 Vega2[15]芯片。下面将以 UltraSPARC 2 核芯片和 AMD Opteron 4 核芯片为例对目前存在的同构多核处理器的测试控制和测试访问机制进行介绍。

3.1 UltraSPARC 2 核芯片测试控制和测试访问机制

UltraSPARC 2 核的芯片是最早研究多核并行测试的处理器。为了达到向后兼容已有的处理器核的可测试性设计[3]技术和支持后续的更多核的处理器设计，文章[14]在 UltrSPARC 2 核芯片上，给出了一种可扩展多核处理器的可测试性设计设计。可扩展的可测试性设计可以支持比双核更多的多核处理器测试，一方面既支持单个核单独进行测试以方便测试结果的分析及诊断，另一方面又可以支持多核并行测试来降低测试时间。

在 UltraSPARC 的 2 核芯片中利用 IEEE1149.1 的测试接口对整个芯片进行测试控制和测试访问。为了方便对每个核进行测试控制，使用了一种层次化的符合 IEEE1149.1 标准的测试访问端口（TAP, Test Access Port），如图 3 所示。芯片内有一个顶层的测试访问端口（Chip-Level TAP），而每个核内还有一个模块级测试访问端口（Block TAP）。 每个测试访问端口都包含标准的 5 个端口 TCK、TMS、TDI、TDO、TRST。顶层的测试访问端口控制整个芯片的边界扫描链、以及芯片的标识符等数据存储器，并完全符合 IEEE1149.1 标准。

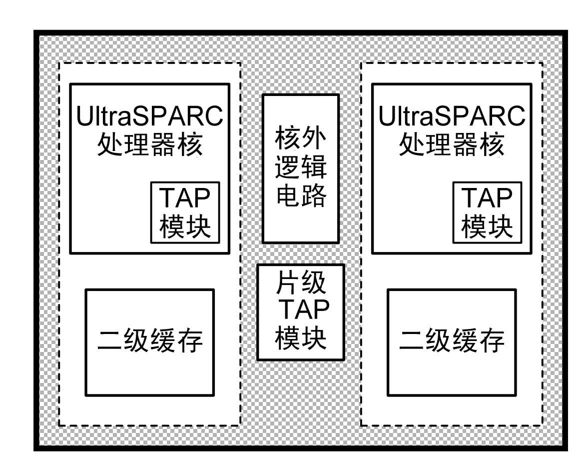


图3. UltraSPARC 的 2 核芯片的 TAP

在 UltraSPARC 2 核芯片中定义了一种锁步（lockstep）模式，在锁步模式下可以达到对多个核进行扫描测试或对多个 MBIST[4]进行测试，并进行测试结果的片上比较。图 4 给出了这种锁步扫描的结构。UltraSPARC 芯片被分为 3 部分：两个内核（Core0、Core1）和顶层的非核内逻辑（non-core）。在图.4 中，`C0_sel` 和 `C1_sel` 控制是否测试 Core0 和 Core1。当为 0 时当前的内核被旁路，而当为 1 时当前的内核被测试，当前内核扫描输入为前一个内核的扫描输出或者是读写的直接输入。这种利用选择器（MUX）网络的扫描连接方法可以保证对芯片中的某一个部分或者某些部分进行测试。当处于锁步模式时，两个核的扫描输入都直接来自读写的输入，并用异或门进行片上比较，选择输出某一个核的扫描数据并给出两个核扫描的比较结果。

锁步模式虽然可以方便地对单个核单独测试和对多个核并行测试，但由于采用了使用多路复用（MUX）网络把扫描输入直接连接到每个核的扫描输入端口和把每个核的扫描输出直接选择输出到读写端口的方案，在核的数目增加（例如 16 核或更多）时，由多路复用网络带来的大量全局连线会造成布局布线的拥塞，从而导致物理设计的困难。

[3] Design for Test, DFT
[4] Memory built-in self-test，存储器内建自测试

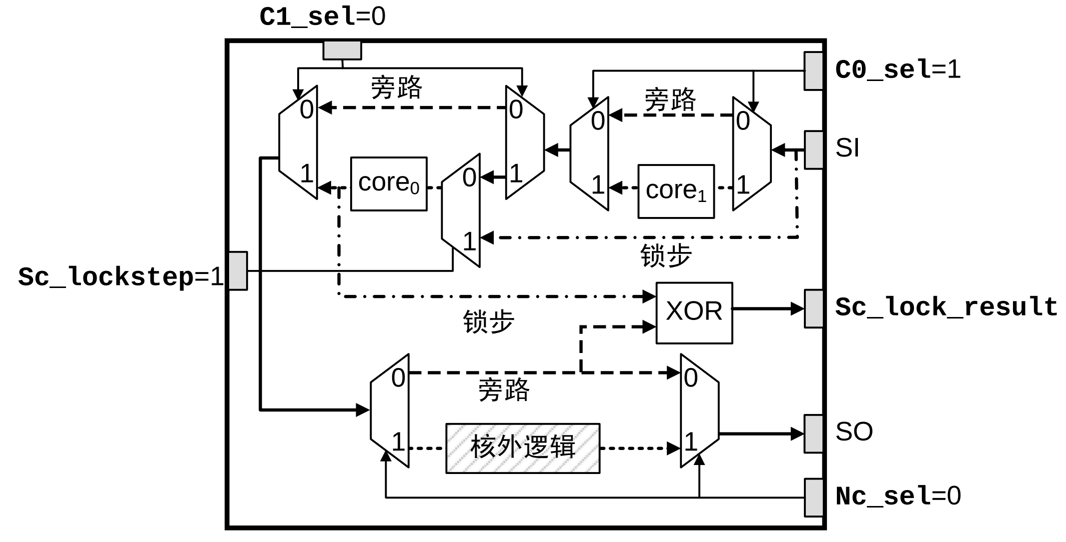


图4. UltraSPARC 的 2 核锁步结构

3.2 AMD Opteron 4 核芯片测试控制和测试访问机制

AMD Opteron 芯片采用了完全符合 IEEE1149.1 标准的一种主从（Master-Slave）模式的测试访问端口结构。主测试访问端口结构放在北桥（Northbridge）模块，并直接连接到顶层标准的 TCK、TMS、TDI、TRSTn 和 TDO 端口上。而从测试访问端口结构放在每个 HT、DDR 和每个处理器核（包括 2 级缓存（Cache））中，如图 5 所示。

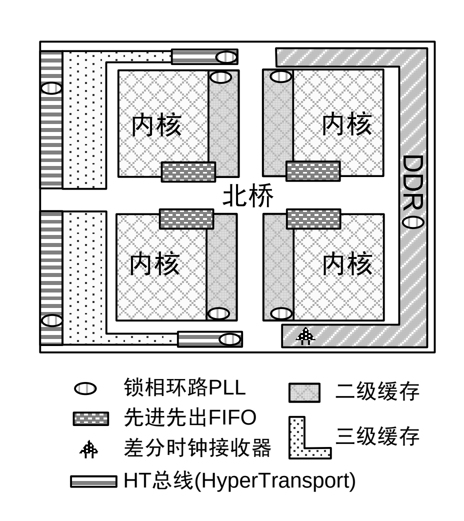


图5. AMD Opteron4 核的结构

主测试访问端口发送给各个从测试访问端口相关的控制信号 Tck、Tms、Tdi、Tdo、Trst、和 TapEn 来控制各个从测试访问端口的操作。其中较为特殊的是 TapEn 信号，这个信号来自于顶层测试访问端口内的配置触发器（Configuration Register），用来控制是否启用模块内的从测试访问端口。当某个模块的 TapEn 使能有效时，此模块内部的边界扫描触发器和指令触发器与顶层边界扫描触发器和指令触发器顺序连接起来。当所有的 TapEn 无效时，TDI 直接广播到各个从测试访问端口控制器上。由于有些模块可能直接和顶层的读写相关，所以在这些模块内直接嵌入了边界扫描触发器，而不是如 UltraSPARC 2 核芯片所设计的把所有边界扫描触发器都集中到顶层。对于这些有边界扫描触发器的模块，TCK、TMS 和 TRSTn 可以直接广播到这些模块的测试访问端口控制器上，以方便对边界扫描触发器的直接控制。

这种主从测试访问端口控制结构可以把内部的每个模块作为一个独立整体进行测试。每个内部测试访问端口既支持 IEEE1149.1 的 4 条公共指令，也支持对内部每个模块进行私有（Private）指令的测试。另一方面通过主测试访问端口控制器对从测试访问端口控制器的控制可以完成对内部每个模块进行公共指令的测试。而且只通过 IEEE1149.1 的标准的几个端口进行测试控制，降低了模块间测试控制连线的复杂度。

扫描链的访问机制是 AMD Opetron 芯片的一个最具特色的地方。为了提高多核处理器

的扫描速度，采用了一种流水的扫描输入和扫描输出结构，如图 6 所示。这种流水的扫描结构不会随着核的增加导致扫描速度的降低。图 6 中 SDI 为扫描输入，SDO 为扫描输出，Cmd 为各种测试指令，例如 Shift、Capture、Mask 以及 Nop 等。这些输入端口都采用存储器锁存的流水结构送入到各个内嵌核的端口。在同一时刻每个核接收的数据是不同的，但当前核当前节拍所接收的数据和前一个核前一节拍所接收的数据相同。利用这种特点每个同构核当前节拍的输出可以和前一个核的输出延迟一节拍的结果进行对比，实现同构核之间的片上实时比较。由于扫描链的数据会存在一定的未知态（X 态），这些 X 态在测试时是属于不确定的数据（Unknown bits），即使对于同构核，这些 X 位也有可能不同。所以这些位不能够进行片上比较。这就需要输入向量屏蔽位（Mask）。SDI 也可以用来输入 Mask 位，当输入 Mask 位时，整个核处于指令空闲（Nop）状态，使 SDI 输入的 Mask 数据存入芯片内的 Mask 存储器中。

图6. AMD Opteron 4 核处理器中测试访问机制详细结构

这种测试控制结构虽然可以完成对同构核同时进行测试，但测试控制非常复杂，需要制定复杂的测试命令序列。而且需要根据测试的数据实时更新每个模块的时钟控制和扫描使能控制，来达到对不同核在不同时刻开启扫描或者捕获时钟的目的。目前常用的电子设计自动化（EDA）工具并不支持对这种结构生成测试向量，需要人工转换和进行验证。这种设计的复杂性会造成大规模多核处理器的测试困难，需要花费较长的测试调试时间，继而会造成测试成本的增长。

4 基于核簇的可扩展的测试访问端口控制器的设计

随着片上核数的增多，对于基于核的测试控制越来越复杂，简单地通过读写端口来控制芯片的测试方案已经变得不再可行。我们基于符合 IEEE1149.1 标准的测试访问端口控制器提出一种多层次的测试控制结构。传统的两层测试控制结构（例如主从结构）在内嵌核数成倍数增加时，会使得主从制器之间连线变长而且贯穿整个芯片，造成物理设计的困难。而这种多层次的测试控制结构局域性强，可以有效地把测试控制信号的传输限制在一个物理设计区域内，从而降低测试传输的时延和物理设计的复杂度。

图 7 给出了一个多层次的测试访问端口控制器结构。在多核芯片的设计中，如果核的个数过多，基于单独的核进行设计，可能造成逻辑和物理设计过于复杂。为了节省设计时间，一般把几个逻辑关系相近的核作为一个核簇，芯片内部可以有多个核簇。随着读写个数的限制，并不可能给每个核都分配固定的测试通道，但可以给同一个核簇内所有的核分配固定的测试通道，使得同一个核簇内的核共享临近的测试通道。多层次的控制结构由测试控制单元（TCU，Test Control Unit）、核簇控制单元（CCU，Cluster Control Unit）和核控制单元（BCU，

Block Control Unit）组成。测试控制单元包含一个符合 IEEE1149.1 的测试访问端口控制器，支持最基本的 EXTEST、PRELOAD/SAMPLE 和 BYPASS 指令，核簇控制单元为核簇的测试控制器，包括一个核簇指令寄存器和一个核簇指令译码器。核控制单元 BCU 包含一个核测试指令寄存器和一个核测试指令译码器。测试控制单元、核簇控制单元的指令寄存器顺序连接，并可以通过 TDI 和 TDO 端口顺序进行访问，而核控制单元的指令寄存器是核簇控制单元指令寄存器的一部分的复制，通过流水的方法顺序传到各个核中，并不直接通过 TDI 和 TDO 进行访问。下面将以图 7 为例对这三个部分分别进行描述。

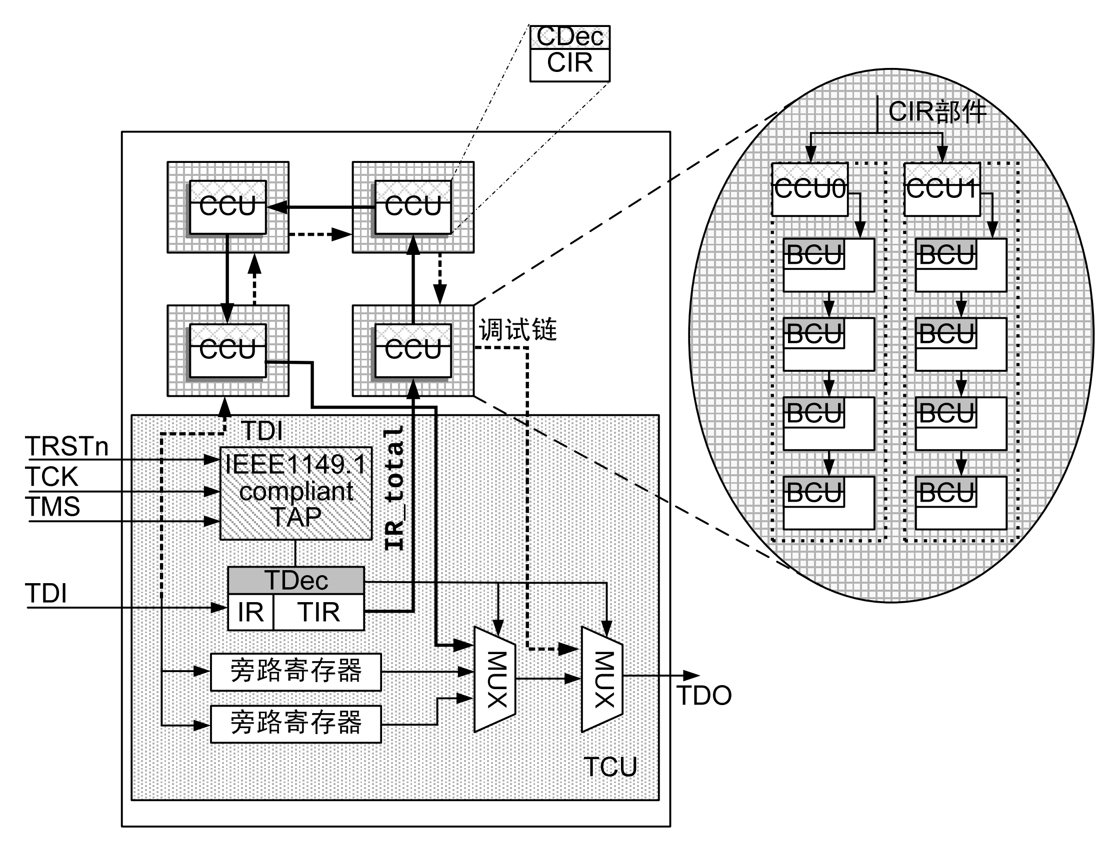


图7. 可扩展的基于核簇的多层次测试访问端口控制器

测试控制单元中的测试访问端口控制器除了可以用来访问旁路寄存器（BYPASS REG）、边界扫描存储器（BSCAN REG）外，还可以用来访问指令寄存器（IR）、测试指令寄存器（TIR）和各个核簇控制单元内部的指令寄存器（CIR）顺序连接而成的 IR_total。其中 IR 存储器存储基本的 IEEE1149.1 指令。当测试相应的核簇时，TIR 内部需要移入相应的核簇的标识符（CID），并由 TDec 解码器输出相应核簇的使能信号并送给核簇控制单元。每个核簇可以单独进行测试，而且由于核簇之间不共享测试通道，几个核簇可以任意组合同时进行测试。当整个芯片处于功能模式时，IR_total 内的所有触发器都为 1，使所有的核簇控制单元使能信号无效，处于旁路指令（bypass）状态。芯片中还设计了一种调试模式，处于该模式时，芯片中所有的触发器连成一条调试链（debug Chain），通过 TDI 和 TDO 端口进行访问。这个调试链既可以用在预烧（Burn-In）和老化测试上（由于测试仪的限制可观测的管脚很少），也可以用在芯片正常功能的在线调试。当芯片功能运行出现问题时，可以通过时钟控制，停止芯片运行，并通过 JTAG 端口扫描出芯片内部所有触发器的状态。其中某些关键触发器的值对分析和诊断芯片出现的故障非常有意义。

核簇控制单元是核簇测试控制模块，顶层的核簇控制单元共享同一组测试通道。核簇控制单元内部还可以继续分级，分为子核簇控制单元，例如分为 CCU0、CCU1、CCU2、

CCU3…CCUn 等。同一层次的核簇控制单元可以共享本核簇控制单元内部所有的测试通道，也可以采用固定通道的方式对核簇控制单元的所有测试通道进行再次分配。具体采用哪种方法取决于测试通道的数目和内部核的个数以及核测试通道的需求。最底层的 CCUn 控制多个被测试核，这些测试核使用同一组测试通道进行测试。核簇控制单元中的内部指令寄存器 CIR 存储子核簇控制单元的标识符（SCID），并通过 CDec 进行解码给出各个子核簇控制单元的使能信号。核簇控制单元内部的指令寄存器还存储核的标识符（COREID）和核簇测试指令，这个标识符作为 CIR 的一部分（Parts of CIR），和每个核的测试指令一起流水地传输到每个被测试核的核控制单元中。

核控制单元是核测试控制模块，每个核控制单元的指令寄存器 BIR 只存储相应的核的标识符和相应核的测试指令，通过 BDec 解码器来确定是否启动对相应核进行测试和进行何种测试。由于测试通道的限制，处于最底层的同一个核簇内的所有的核只能进行同一种测试。例如同一底层核簇内的核不能有些核进行扫描测试，而另外一些核进行 MBIST 的测试。虽然这在某种情况下限制了测试组合的自由程度，但在另一方面降低了同一个核簇内测试通道共享机制设计的复杂度。

通过测试控制单元、核簇控制单元和核控制单元的控制，可以完成对同一核簇内的核进行单独测试或者和不同核簇内的核同时进行测试。同一核簇内的核由于共享同一组测试通道，对于异构核只能单独进行测试，而对于同构核通过一种同构核片上比较机制可以完成同构核的并行测试，这部分的设计将在下一节进行描述。而对于 MBIST 来说，完全可以通过核的测试指令和测试模式来实现多个核内的所有 MBIST 同时测试或单个核内的所有 MBIST 同时进行测试，相对于扫描测试来说实现相对简单。

5　同构核扫描测试访问机制的设计

针对同构多核处理器的特点，我们设计了一种简单易行的带有片上比较的扫描测试访问机制，采用同一组测试向量来完成对所有同构核的测试。图 8 给出了基于四个同构核的扫描访问和比较机制的结构图。

由于每个核的物理位置不同，扫描测试输入向量是通过流水的方式传播到每个同构核的，而且同构核的顺序可以根据物理设计位置的需求进行任意调换。同时为了保证每个 IP 核在同一时刻接收到相同的数据，每个核都包含了不同数目的扫描输入平衡寄存器（Scan Input Balance Registers, SIBR）。

扫描输出只输出被选中核的扫描测试结果，其他同构核的扫描输出和被选中核的扫描输出做片上比较。选中核的扫描输出同样也是通过流水的方式传递到下游的每个同构核。为了保证每个同构核的扫描输出的时刻相同，每个同构核后插入了扫描输出平衡寄存器（Scan Output Balance Register, SOBR）。

由于很多宏单元（macro cell）（比如处理器核中的缓存存储器）的存在，电路中不可避免地会有很多不确定状态的点，即这些点的值是未知的，而这会导致向量产生 X 位。另一方面，检测故障有时候不需要对所有的触发器和原始输入（PI）进行赋值，也不需要对所有的触发器和原始输出（PO）进行观测。这些不需要观测的点在向量响应中也表现为 X 值。另外，向量生成过程中用户施加的一些和时序相关的限制也会导致向量响应中出现 X 值。X 的存在会破坏同构核之间的比较结果，引起比较结果的错误。为了避免向量中 X 的存在所导致的比较错误，我们采用了额外的输入管脚输入屏蔽信息（Mask）到每个比较器，使得比较器只对确定值（0 或 1）进行比较。需要的额外管脚数目等于扫描输出管脚的数目。在

含有压缩结构的电路中，可以采用非对称（asymmetrical）的压缩结构来减少扫描输出管脚的数目，从而减少额外的输入管脚。为了降低对时序的要求，扫描屏蔽信息也是通过流水的方式传播到每个同构核。

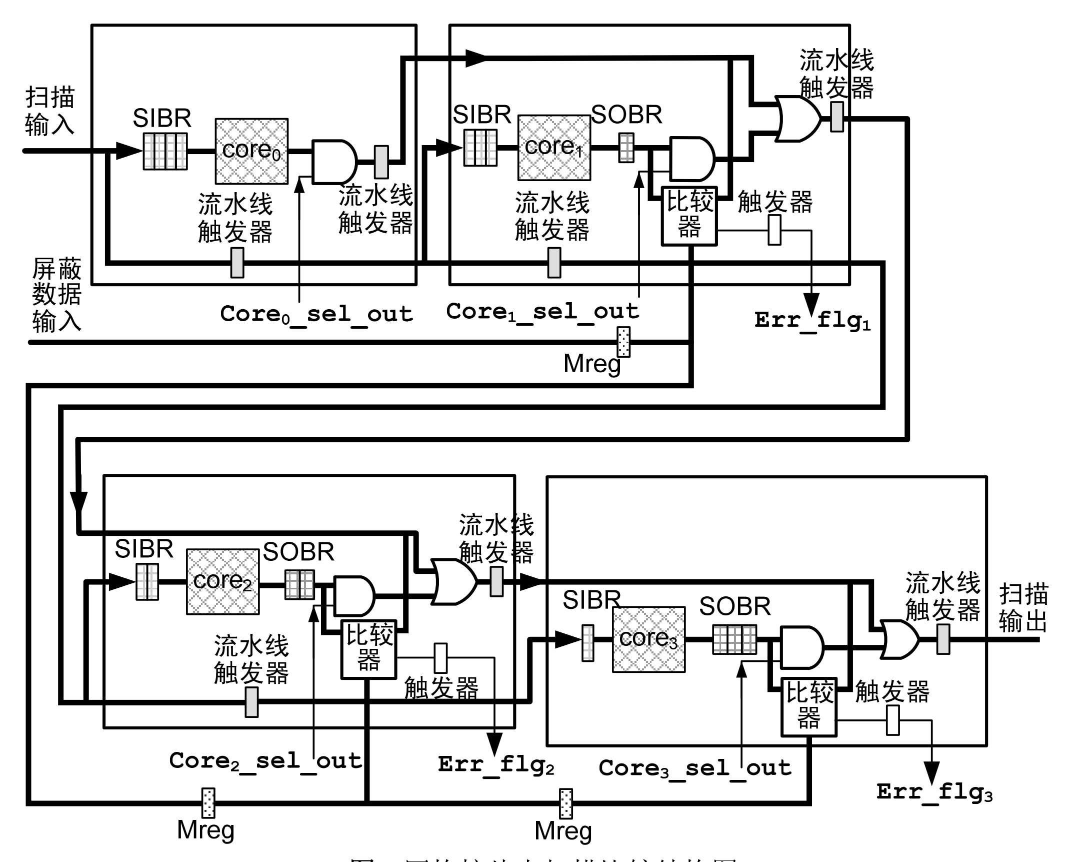


图8. 同构核片上扫描比较结构图

下面根据图 8 详细描述所设计的扫描测试访问机制。

扫描输入数据（Scan in）依次通过流水触发器（Pipeline Reg）进入不同的内核 $Core_0$、$Core_1$、$Core_2$ 和 $Core_3$ 的 SIBR 中。屏蔽数据（mask data）也依次经过流水触发器 Mreg 进入 $Core_0$、$Core_1$、$Core_2$ 和 $Core_3$。而测试响应数据首先通过 SOBR 然后再通过流水触发器按照这个顺序依次经过 $Core_1$、$Core_2$ 和 $Core_3$ 传送到扫描输出端口（Scan out）。$Core_0$ 有 4 个 SIBR 和 0 个 SOBR，$Core_1$ 有 3 个 SIBR 和 1 个 SOBR，$Core_2$ 有 2 个 SIBR 和 2 个 SOBR，$Core_3$ 有 1 个 SIBR 和 3 个 SOBR。这些输入和输出的平衡触发器和 4 个扫描输入输出的流水触发器相配合，使得这四个核都能够在延迟 4 个节拍后得到扫描输入数据，在 4 个节拍后输出扫描输出数据，同时保证了 4 个核能够同时进入移位状态和捕获状态。

屏蔽信息（mask data）从向量文件中抽取。如果向量响应中某位为确定值 0 或 1，则相应的 mask 位为 0；如果响应值为 X，即不进行观测，则相应的 mask 位为 1。这样就能屏蔽掉 X 的影响，保证比较结果的正确性。由于基准核的测试响应数据到达每个核的时间不一样，所以每个核开始比较的时间也不一样。如果 $Core_0$ 直接被观测，则 $Core_1$ 首先开始比较操作，$Core_{(n-1)}$最后开始比较操作。一般 $Core_m$ 是在 m 个时钟周期后开始比较操作的，这就要求在前 m 个时钟周期，传播到 $Core_m$ 的屏蔽位必须为 1。从屏蔽信息输入端口来输入这些 mask 位是不可行的，因为屏蔽信息到达 $Core_m$ 也需要 m 个时钟周期。我们采用特殊的流水寄存器 Mreg 来解决这个问题。这些特殊的 Mreg 寄存器能在扫描准备阶段（`test_setup`）和扫描捕获阶段（`capture`）过程中捕获“1”值。这样，对于 $Core_m$，在前 m 个时钟周期，

保存在 Mreg 寄存器中的“1”用来屏蔽掉相应的比较，m 个时钟周期后，来自屏蔽信息输入端口的数据用来屏蔽掉对 X 值的比较。

除了最上游的 $Core_0$ 之外，其它的核都包含一个比较器，用以比较该核的响应与来自上游核的响应，并给出 Err_flg 信号。

如果对四个同构核一块进行测试，则 $Core_0$ 应首先被直接观测，它的响应作为基准响应与其它核进行比较, 即控制 Core0_sel_out=1, Core1_sel_out=0, Core2_sel_out=0，Core3_sel_out=0。ATE 设备对期望响应与 $Core_0$ 的响应进行比较，判断 $Core_0$ 是否存在故障，同时 Err_flg(1~3)分别给出 $Core_1$、$Core_2$、$Core_3$ 与 $Core_0$ 进行比较的结果。如果 $Core_0$ 的响应与期望响应完全一致，则说明 $Core_0$ 不存在故障，那么 Err_flg(1~3)就反映了 $Core_{(1\sim3)}$ 是否存在故障，测试过程结束；如果 $Core_0$ 的响应与期望响应不一致，则说明 $Core_0$ 存在故障，那么 Err_flg(1~3)给出的信息就不具有参考性，只是说明了与 $Core_0$ 的结果是否一致，这时还需要新一轮的测试。当 $Core_0$ 存在故障时，需要将 $Core_0$ 旁路掉，直接输出 $Core_1$ 的结果，$Core_2$、$Core_3$ 与 $Core_1$ 进行比较来判断其正确性，即控制 Core0_sel_out=0, Core1_sel_out=1, Core2_sel_out=0,Core3_sel_out=0。这时，Err_flg1 的输出没有意义，因为 $Core_1$ 的响应是在测试仪上进行比较的，不是片上比较的。当 $Core_1$ 不存在故障时，$Core_2$ 和 $Core_3$ 是否存在故障则分别由 Err_flg2 和 Err_flg3 给出，测试过程结束；否则还需要进一步的测试来判定 $Core_2$ 和 $Core_3$ 是否存在故障。依此类推，直到所有的核是否存在故障都被判定。可以看出，最好情况下（$Core_0$ 不存在故障的情况），只需要一轮测试就可以对所有的同构核进行判定；最坏情况下（$Core_0$，$Core_1$ 和 $Core_2$ 都存在故障的情况），需要 4 轮测试才能对所有的核进行判定。

某些情况下，基于功耗的考虑，一般不同时对所有的核进行测试。比如，在圆片测试阶段（wafer test），圆片未进行切割，所有的芯片是紧挨着，这种情况下散热效果不好，如果所有的核一起测试，极易因功耗过大而造成对芯片的损害。在对功耗要求比较严格的情况下，可以选择少数核进行同时测试。比如一次可以选择两个核进行测试或者一次只选择一个核进行测试。

6 总结

片上多核处理器已经成为微处理器设计的一个主流方向。片上多核处理器的规模及复杂性的急剧增加，给片上多核处理器的测试设计带来了巨大的挑战。一方面，晶体管规模的急剧增长造成测试向量、测试时间和测试开销的急剧增长。另一方面，虽然多核处理器芯片的规模不断地增加，但芯片的读写端口数和单核处理器相比基本保持不变。如何利用有限的测试通道资源对大规模处理器进行测试是多核处理器测试面临的一个重要问题。本文提出一种采用了多层次的基于核簇的测试控制方法，能够根据读写的物理位置合理地分配给每个核簇相应的测试通道，并采用流水的方法把测试数据传到各个模块中，大大降低了时序的要求和物理设计的难度。而且为了降低测试开销，采用了相邻核间的比较机制。这种技术既避免了复杂的控制指令的输入，又可实时地输入屏蔽位，同时还能使用商业的 ATPG 工具生成测试向量，大大降低测试设计的复杂度，提高了应用的便利性。

参考文献：

[1] The International Technology Roadmap for Semiconductors(ITRS), Semiconductor Industry Association, 2007 Edition.

[2] IEEE Standard, “Testability Method for Embedded Core-based Integrated Circuits”, IEEE std.1500,

2005.

[3] J. Aerts, E.J. Marinissen, “Scan Chain Design for Test Time Reduction in Core-based ICs”, Proc. International Test Conference, pp.448-457, 1998.

[4] P. Varma, S. Bhatia, “A Structured Test Re-Use Methodology for Core-Based System Chips”, Proc. International Test Conference, pp.294-302, 1998.

[5] E. J. Marinissen, R. Arendsen, G. Bos, H. Dingemanse, M. Lousberg, C. E. Wouters, “A Structured and Scalable Mechanism for Test Access to Embeded Reusable Cores”, *Proc*. International Test Conference, pp.284-294, 1998.

[6] I. Ghosh, S. Dey, N. K. Jha, “A Fast and Low Cost Testing Techniques for Core-based System-on-Chip”, Proc. ACM/IEEE Design Automation conference, pp.542-547, 1998.

[7] N. Touba, B. Pouya, “Using Partial Isolation Rings to Test Core-Based Design”, *IEEE Design & Test of Computers*, Vol. 14, No.2, pp.52-59, 1997.

[8] P. Harrod, “Testing Reusable IP-A Case Study”, Proc. International Test Conference, pp.493-498, 1999.

[9] M. Benabdenbi, W. Maroufi, M. Marzouki, “CAS-BUS: A Scalable and Reconfigurable Test Access Mechanism for System on a Chip”, Proc. Design, Automation and Test in Europe, pp.141-15, 2000.

[10] M. Riley, L. Bushard, N. Chelstrom, N. Kiryu, S. Ferguson, “Testability Features of the First-generation CELL Processor”, Proc. International Test Conference, pp.111-119, 2006.

[11] P. J. Tan, T. Le, K. H. Ng, P. Mantri, J. Westfall, “Testing of UltraSPARC T1 Microprocessor and its Challenges”, Proc. International Test Conference, paper 16.1, 2006.

[12] R. Molyneaux, T. Ziaja, H. Kim, S. Aryani, S. Hwang and A. Hsieh, “Design for Testability Features of the SUN Microsystems Niagara2 CMP/CMT SPARC Chip”, Proc. International Test Conference, paper 1.2, 2007.

[13] T. Wood, G. Giles, C. Kiszely, M. Schuessler, D. Toneva, J. Irby, M. Mateja, “The Test Features of the Quad-Core AMD Opteron Microprocessor”, Proc. International Test Conference, paper 2.2, 2008.

[14] I. Parulkar, T. Ziaja, R. Pendukar, A. D’Souza and A. Majumdar, “A Scalable, Low-cost DFT architecture for UltraSPARC Chip Multiprocessors”, Proc. International Test Conference, pp.726-735, 2002.

[15] S. Makar, T. Altinis, N. Patkar and J. Wu, “Testing of Vega2, a chip multi-processor with multiple cores”, Proc. International Test Conference, paper 9.1, 2007.

作者简介：

齐子初： 中国科学院计算技术研究所微处理器中心，高级工程师，qizichu@ict.ac.cn

刘　慧： 中国科学院计算技术研究所微处理器中心，工程师